Kthmax. K-ый максимум

Имя входного файла: `kthmax.in`
Имя выходного файла: `kthmax.out`

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k-й максимум.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество команд ($n \le 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \le 10^9$). Поддерживаемые команды:

- $+1$: Добавить элемент с ключом k_i.
- 0: Найти и вывести k_i-й максимум.
- -1: Удалить элемент с ключом k_i.

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i-го максимума, он существует.

Формат выходного файла

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i-й максимум.

Пример

kthmax.in	kthmax.out
8 +1 5 +1 3 +1 7 0 2 -1 3 0 1 1 10 0 3	5 7 5

Firesafe. Противопожарная безопасность

Имя входного файла: `firesafe.in`
Имя выходного файла: `firesafe.out`

В городе Судиславль n домов. Некоторые из них соединены дорогами с односторонним движением.

В последнее время в Судиславле участились случаи пожаров. В связи с этим жители решили построить в городе несколько пожарных станций. Но возникла проблема — едущая по вызову пожарная машина, конечно, может игнорировать направление движения текущей дороги, однако, возвращающаяся с задания машина обязана следовать правилам дорожного движения (жители Судиславля свято чтут эти правила!).

Ясно, что где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой выехала. Но строительство станций стоит больших денег, поэтому на совете города было решено построить минимальное количество станций таким образом, чтобы это условие выполнялось. Кроме того, для экономии было решено строить станции в виде пристроек к уже существующим домам.

Ваша задача — написать программу, рассчитывающую оптимальное положение станций.

Формат входного файла

В первой строке входного файла задано число n ($1 \le n \le 3\,000$). Во второй строке записано количество дорог m ($1 \le m \le 100\,000$). Далее следует описание дорог в формате a_i b_i, означающее, что по i-й дороге разрешается движение автотранспорта от дома a_i к дому b_i ($1 \le a_i, b_i \le n$).

Формат выходного файла

В первой строке выведите минимальное количество пожарных станций K, которые необходимо построить. Во второй строке выведите K чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

Пример

firesafe.in	firesafe.out
5 7 1 2 2 3 3 1 2 1 2 3 3 4 2 5	2 4 5

Diameter. Диаметр дерева (*)

Имя входного файла:	`diameter.in`
Имя выходного файла:	`diameter.out`

Расстоянием между двумя вершинами в неориентированном невзвешенном графе называется количество ребер в кратчайшем пути, соединяющем эти вершины.

Диаметром графа называется максимальное расстояние между вершинами этого графа.

Деревом называется связный граф без циклов.

Дано дерево, требуется найти его диаметр.

Формат входного файла

Первая строка входного файла содержит натуральное число n — количество вершин в дереве ($n \le 100\,000$). Следующая строка содержит n чисел, i-е число — номер вершины, являющейся предком вершины номер i (если вершина является корнем, то соответствующее значение равно нулю).

Формат выходного файла

Первая строка выходного файла должна содержать одно число — диаметр данного дерева.

Пример

diameter.in	diameter.out
3 2 0 2	2
7 5 4 0 3 4 7 3	5